SMART HOME SECURITY AND AUTOMATION SYSTEM USING IOT

A Mini Project Report

submitted by

DEEPIKA BALAKRISHNAN C (MES20MCA-2015)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

February 2022

DECLARATION

I undersigned hereby declare that the project report **SMART HOME SECURITY AND AUTOMATION SYSTEM USING IOT**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by me under supervision of Dr.Geevar C Zacharias , Assistant Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: KUTTIPPURAM

DEEPIKA BALAKRISHNAN C (MES20MCA-2015)

Date: 04/03/2022

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **SMART HOME SECURITY AND AUTOMATION SYSTEM USING IOT** is a bona fide record of the Mini Project work carried out by **DEEPIKA BALAKRISHNAN C (MES20MCA-2015)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

Head Of The Department

Acknowledgements

First of all, I would like to give thanks to God for giving me the opportunity to complete and submit this project. Without the help and blessings of God the Almighty, it would not have been possible to accomplish this project. With great respect I express my sincere thanks to Dr.K.A Navas, Principal, MES College of Engineering, providing facilities for this project. I would also like to express my gratitude towards my supportive and encouraging project co-ordinator, Ms.Priya J.D Assistant professor, Department of Master of Computer Applications, for allowing me to do this project and for her inspiring guidance, reliability, constructive criticism and challenging but motivational feedback throughout the course of this project. I would like to express my sincere thanks to Dr.Geevar C Zacharias Assistant professor, Department of Master of Computer Applications, my internal guide and all of the faculty members of the Department of Master of Computer Applications for equipping me with the skills and knowledge of our field of education required to accomplish this project. Your contribution will doubtlessly be acknowledged and I will always remain grateful to you all. Last but not least my graceful thanks to my parents, friends and also the persons who supported me directly and indirectly during the project.

DEEPIKA BALAKRISHNAN C (MES20MCA-2015)

Abstract

Internet of Things is a system where appliances are embedded with software,sensors and actuators. Wi-Fi is one of the main wireless communication protocols for connecting different devices for exchange of data over Internet. IoT is implemented in smart home security to device embedded module for standalone operation of collecting and monitoring different sensor data for home security. This project focuses on building a home security system which will be wireless. Security over a network is achieved using ardunio uno. This system was used for monitoring the status of home by using different sensors.

Keywords:Internet of Things, Home Automation, Smart Security, Camera, Arduino uno

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Homes of the modern world will become smarter and more automated due to the advantages of automation. Automation system in smart home allows users to control different electric and electronic appliances. Safety and security of the home is one of the most primary concerns. The advancement of technology now provides the option of the safety and security of people along with their belongings. This project focuses on a system that provides features of Home Automation relying on Internet of Things (IOT) to operate easily, in addition to that it includes a camera module and provides home security.

Security is achieved with motion sensors if movement is sensed at the entrance of the house; a notification is sent that contains a photo of house entrance in real time. This notification will be received by the owner of the house via internet such that app can trigger a notification. So owner can view the visitor is a familiar person or an unknown person. The user can make use of this system to control switching on of lights, fan, etc. automatically.

The project mainly aims to overcome the shortcomings of home security systems by providing information of current situation when the owner is away from the house.It will also enhance the IOT's network security using encryption and decryption of the user's data.

1.1.1 Motivation

The motivation for developing smart home systems comes from many reasons, but most prominent are convenience, security, energy management, connectivity and luxury. Smart Home systems are one of the newer areas of research that have not been fully integrated into our society. This is because the research requires many other disciplines of research and engineering to produce a functional smart home. The cost of installing a smart home is also a large hindrance to the emergence of smart home systems into the market. The extra cost of the install is from the fact that even though a majority of homes were built in the near past, technology has been growing exponentially. This means that most homes were built before this technology was available, and this creates a barrier for the development and sales of smart home systems. However the technology is becoming better and cheaper, and this will help to make smart home systems an expense worth having when new homes are being built.

1.2 Objective

- Home surveilance and security feature makes house protection high
- From any where we can on or off our home appliances, which provides better usage of devices
- User can get the list of persons who visits home easily

1.3 Report Organization

The project report is divided into four sections.
Section 2 describes literature survey.
Section 3 describes the methodology used for implementing the project.
Section 4 gives the results and discussions.
Finally Section 5 gives the conclusion.

Chapter 2

Literature Survey

Central Controller-based Home Automation System

A central controller-based home security system looks to improve the security of the homes in a locality by combining many homes into a security network with a control node dedicated for each locality depending on the number of users. These control nodes are controlled by a few central or chief control nodes with considerably high processing power. The security system described by S. Tsai et al. [31], called Home Security System on Intelligent Network (HSSIN), uses such a central controller-based approach. The proposed system lacks modern security parameters.

A central controller based security system has its own unique challenges:

- All or most homes in the neighborhood have to join in for the approach to be cost effective and successful.
- The main question we have to consider here is who controls or has access to the central controller and its data? The central controller will be able to know about a home's intimate and private information from the data at its disposal, like if a home's room heater is on, or if an inhabitant in a home is taking a shower. This raises serious privacy concerns. We already know how people feel about government surveillance on the Internet. Central controller-based security systems provide an opportunity to do even more privacy-violating surveillance on homes.

The work of K. Atukorala et al. [32] proposes a home automation Security System called SmartEye using General Packet Radio Service (GPRS). SmartEye also uses a central controller, to which many individual home controllers are connected. The system proposes a real-time home automation and monitoring system. The system alerts the homeowner by mobile phone using GPRS, and the user can view the home using live camera feeds. The system uses a RabbitCore Module to connect an electrical appliance in the home to the home system, usually a PC. Each home system is connected to a central server. RabbitCore has an IP address, so each device connected to it can be identified and operated via mobile phones using GPRS. The user sends device management commands to a central server. The home system reads the command from the central server, called home polling, and makes the changes needed to a device. When a device changes state, the home system, usually a PC, sends the change of state of the device to the central server. The user's mobile will read the change from the central server, called mobile polling. The mobile user is provided with a home plan and places where each piece of equipment is kept in their home. The proposed research gives importance to communication and network setup rather than security. It mentions intrusion detection, but no concrete parameters identifying intrusions are mentioned.

SmartEye uses video cameras for security. Its security issues are discussed below. Moreover, like all centralized home security systems, the proposed system is also not ideal for securing single homes, but suits a group of homes best, and the author's claims of "increase in poll rate leads to increase in security" is debatable and misleading.

In reality, a central controller-based security system is difficult to implement and raises some very serious privacy concerns.

Bluetooth-based Home Automation System

The work of N. Sriskanthan et al. [33] shows the implementation of a home automation system using Bluetooth. They use a host controller implemented on a PC, which is connected to a microcontroller-based sensor and device controllers. The researchers even built a new protocol on top of the Bluetooth software stack, called Home Automation Protocol (HAP), to make the communication between devices possible. The device controller is connected to electronic devices through th e I2C Bus. The system allows more than one device controller to be connected to the host controller.

The work of H. Kanma et al. [34] also proposes a home automation system using Bluetooth that can be accessed remotely through GPRS. The researchers use a cellphone equipped with Bluetooth connectivity as a host controller and a GSM modem that provides Internet connectivity. Home devices are fitted with Bluetooth communication adapters so that they can communicate with the host controller phone via Bluetooth. The paper discusses remotely controlling and updating home devices along with fault diagnostics and detection. The work also talks about providing an electronics user manual on the phone using Bluetooth and Internet.

Issues of using Bluetooth for home automation:

- Bluetooth has a maximum communication range of 100m in ideal conditions. More may be needed in a home environment.
- Bluetooth communication has comparatively high power consumption, so the batteries of devices need to be frequently recharged or replaced.
- Bluetooth technology has advanced and improved to Bluetooth Low Energy (BTLE), which provides the same range of communication. However, it has serious security concerns such as eavesdropping and weak encryption as discussed by M. Ryan [35].
- Bluetooth communication should only be used on occasions where there is a need for quick short-lived network communication with little concern for security.

Bluetooth looks like an attractive communication technology for creating smart homes. It is cheap, easy, and quick to set up. People are already familiar with the technology. The hardware required for establishing

Bluetooth communication is readily available. And the technology also provides the necessary bandwidth for the operation in a home. But they also have serious flaws, as discussed above.

Chapter 3

Methodology

3.1 Introduction

In this project I am developing an android application with embedded system for control devices and home security .This project contains main modules like Users, Desktop application, and embedded system User can view the visitors and control the electrical devices User can register and they can login, they can add familiar persons for face recognition and they can also view the familiar person, edit the familiar person and remove familiar persons. Through camera surveillance can identify visitors and familiar persons using face recognition and user can view visitor logs entered by camera module.User can view the device status by phone and can update device status according to the instructions obtained. Here the device control is done using Embedded system.

Mainly there are two technical sides they were: Face recognition is a method of identifying or verifying the identity of an individual using their face. Recognition of familiar faces involves a match between the products of structural encoding and previously stored structural codes describing the appearance of familiar faces, held in face recognition units. In my project facial recognition is used to help the owners to recognise their familiar persons. I have used dlib, cmake libraries of python in-order to implement face recognition.

Another important feature of the project is Embedded system for control devices

3.2 MODULES

3.2.1 USER

- Login
- User registration
- Change password
- Familiar person management
- View visitor log
- Device control
- Receive Notification

3.2.2 DESKTOP APPLICATION (CAMERA MODE)

- Surveillance
- Visitor log entry
- Face recognition

3.2.3 EMBEDDED

- Device control based on user instruction

3.3 DEVELOPING ENVIRONMENT

3.3.1 HARDWARE REQUIREMENTS

The selection of hardware is very important in the existence and proper working of any software. Then selection hardware, the size and capacity requirements are also important.

- Input Device : Mouse, Keyboard
- Output Device : Monitor
- Memory : 4 Gb Ram (Minimum)
- Processor : Intel core i3 or above

3.3.2 SOFTWARE REQUIREMENTS

One of the most difficult task is selecting software for the system, once the system requirements is found out then we have to determine whether a particular software package fits for those system requirements. The application requirement:

- Operating System : Windows 8 /10for Better Performance , Android
- Front End : Android
- Back End : MySQL
- Software Used : PyCharm , Android Studio , SQLyog

3.4 EMBEDDED SYSTEM

Embedded systems consist of a micro-controller with on-board memory, a power supply, and communication ports for transmitting data to other devices. Embedded software programs tell the micro-controller how to respond in real time to data collected from the environment through peripheral sensors and devices. Embedded systems are used to enhance safety, reduce costs, and offer convenience and cost-savings to customers in industries that include automotive, home/consumer electronics, communications, and healthcare.

3.4.1 Block Diagram

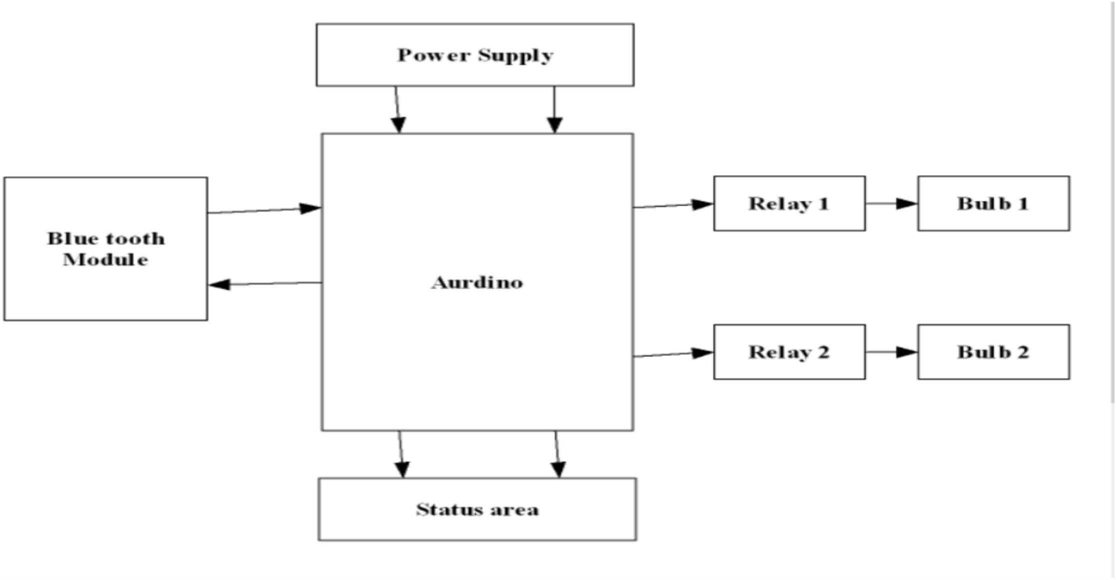


Figure 3.1: Block Diagram

3.4.2 Steps of Block Diagram

In this diagram there is a Power supply, Status area, Arduino, Bluetooth module, 2 relays and 2 bulbs. For the power supply the Arduino, Bluetooth module and relay need 5 volt.12 volt is required to cover the whole project. The Power supply section of the circuit allows to convert the 12 volt adapter into 5 volt Here we are using DC adapter, if we are using AC adapter it convert it into DC adapter having 5 volt. The status area section contains two LED's: status LED and data LED. When a program loop inside the Arduino, the status LED is used to switch the LED to on and off at the starting of a program, it repeats depending on the program.it allow us to understand our program is running well. Status LED is in green color. Data LED is in blue color, this blue color indicates if there is any data transaction is taken place through the Bluetooth module, the LED starts to on and off. Relay is a switching device. The 230 volt power supply is not directly sent to bulb, it is done through passing it across the relay and then to bulb. In order to do that we need to make the relay short we need to apply 5 volt power into the relay from Arduino, then only the power is supplied to bulb, other time it is open. When an app start there is a button to connect the Bluetooth module device. When we click the button it list all the nearby devices that are on, in that list our Bluetooth device will also come if its power is on, then just click the device to connect it. When it is connected, through the Bluetooth device we can sent any data or character from the Arduino broad and it is received through the app. This is the way the app and Arduino is communicated. If we give a command to on the bulb in app, it receives from Bluetooth through Arduino board. Arduino board process it and look the letter if it is A one bulb is on, B next bulb is on, C one bulb is off, D next bulb is off. We also sent the status of the bulb through Arduino to Bluetooth

3.4.3 HC-05 BLUETOOTH MODULE

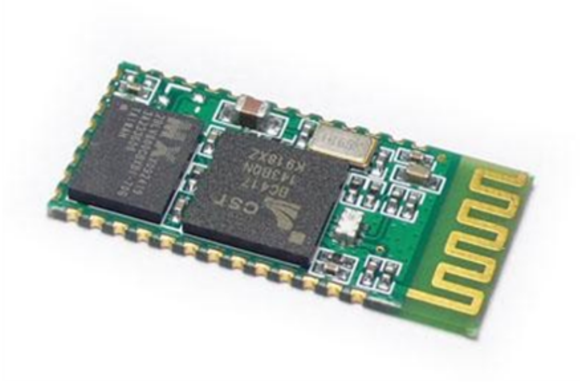


Figure 3.2: HC-05 BLUETOOTH MODULE

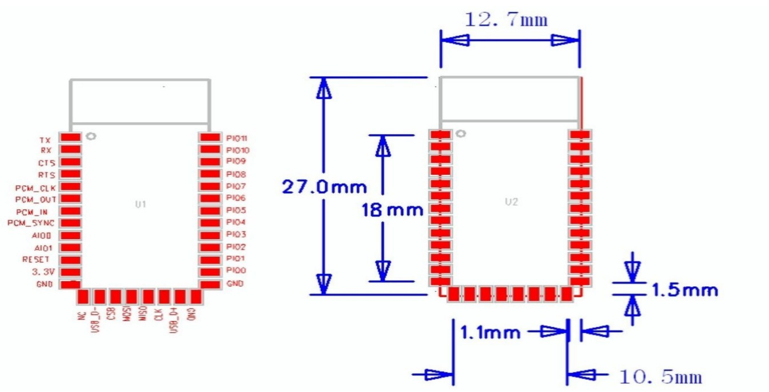


Figure 3.3: Bluetooth Module Hardware

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

Specifications

Hardware features

- -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

Software features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has. Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.

- Permit pairing device to connect as default.
- Auto-pairing PINCODE:”0000” as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

Arduino Uno

Open-source electronic prototyping platform enabling users to create interactive electronic objects.

The Arduino UNO board contains the following components and specifications:

ATmega328: This is the brain of the board in which the program is stored.

Ground Pin:there are several ground pins incorporated on the board.

PWM: the board contains 6 PWM pins. PWM stands for Pulse Width Modulation, using this process we can control the speed of the servo motor, DC motor, and brightness of the LED.

Digital I/O Pins: there are 14 digital (0-13) I/O pins available on the board that can be connected with external electronic components.

Analogue Pins: there are 6 analogue pins integrated on the board. These pins can read the analogue sensor and can convert it into a digital signal.

AREF: It is an Analog Reference Pin used to set an external reference voltage.

Reset Button: This button will reset the code loaded into the board. This button is useful when the board hangs up, pressing this button will take the entire board into an initial state.

USB Interface: This interface is used to connect the board with the computer and to upload the Arduino sketches (Arduino Program is called a Sketch)

DC Power Jack: This is used to power up the board with a power supply.

Power LED: This is a power LED that lights up when the board is connected with the power source.

Micro SD Card: The UNO board supports a micro SD card that allows the board to store more information.

3.3V: This pin is used to supply 3.3V power to your projects.

5V: This pin is used to supply 5V power to your projects.

VIN: It is the input voltage applied to the UNO board.

Voltage Regulator: The voltage regulator controls the voltage that goes into the board.

SPI: The SPI stands for Serial Peripheral Interface. Four Pins 10(SS), 11(MOSI), 12(MISO), 13(SCK) are used for this communication.

TX/RX: Pins TX and RX are used for serial communication. The TX is a transmit pin used to transmit the serial data while RX is a receive pin used to receive serial data.

3.5 FACE DETECTION

Face recognition is a method of identifying or verifying the identity of an individual using their face. Recognition of familiar faces involves a match between the products of structural encoding and previously stored structural codes describing the appearance of familiar faces, held in face recognition units. In my project facial recognition is used to help the owners to recognise their familiar persons. I have used dlib, cmake libraries of python in-order to implement face recognition.

3.5.1 CMake

CMake is an open-source, cross-platform family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files, and generate native makefiles and workspaces that can be used in the compiler environment of your choice. The suite of CMake tools were created by Kitware in response to the need for a powerful, cross-platform build environment for open-source projects such as ITK and VTK.

3.5.2 Dlib

Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. Dlib's open source licensing allows you to use it in any application, free of charge.

In this system, On motion detection camera captures an image of the person in front of the door, Then real-time face recognition is done using Dlib and CMake libraries. If person's image matches one of the prestored familiar person's image then the system sent a notification to the owner that it's a familiar person. Otherwise sends the notification that it's strange person along with his/her's image.

3.6 AGILE METHODOLOGY

The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins,teams cycle through a process of planning,executing and evaluating. Continuous collaboration is vital.

This project is divided into 4 sprints:

Sprint 1

-Table design

-Form design

-Coding

Sprint 2

-Familiar person management

-Visitor log management

Sprint 3

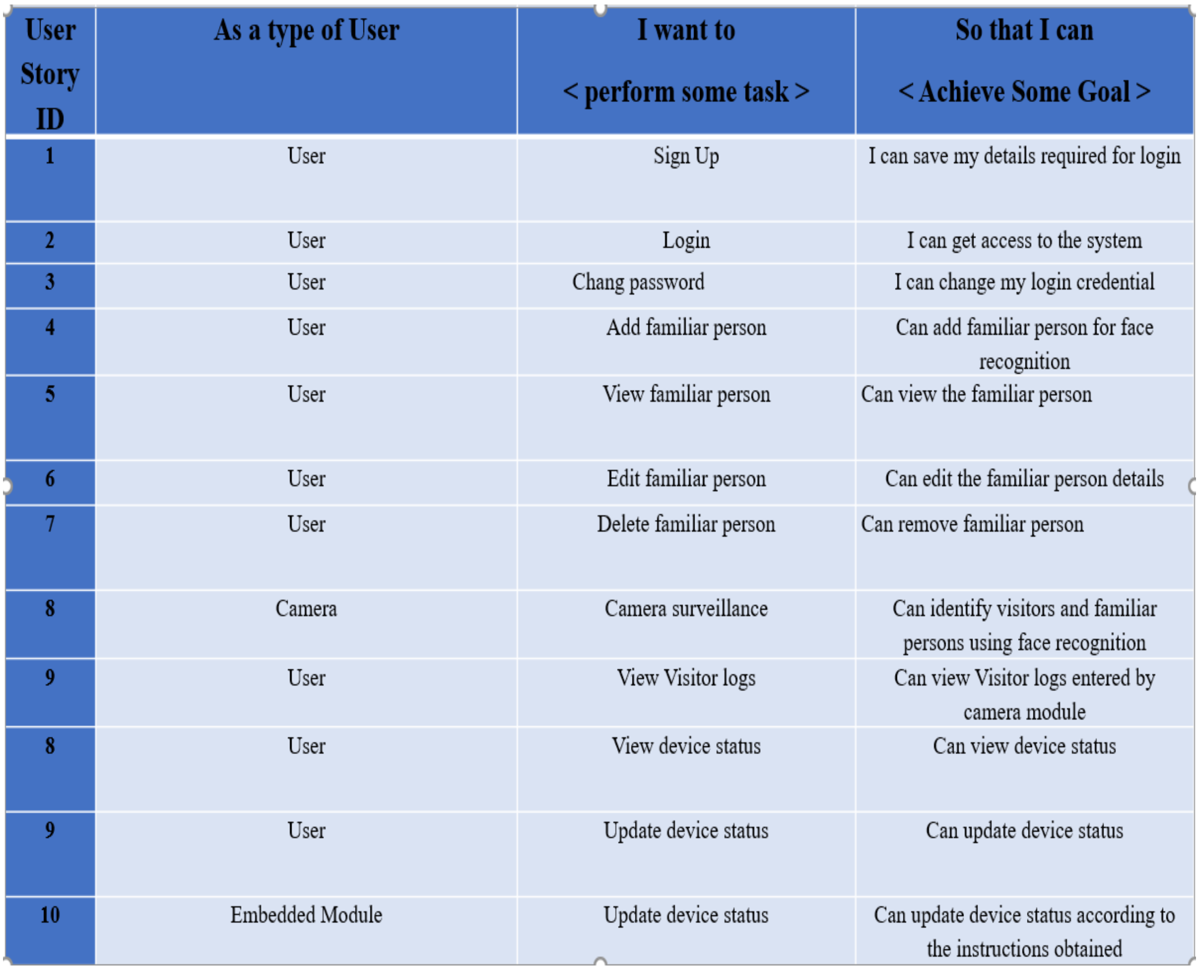
-Embedded system

Sprint 4

-Testing data

-Output generation

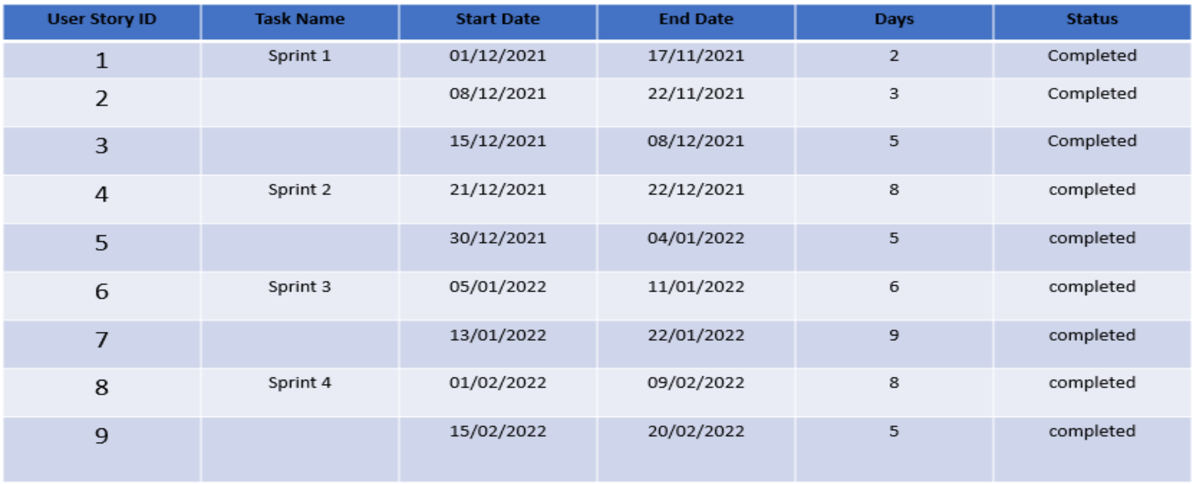
3.6.1 USER STORY



User Story ID	As a type of User	I want to < perform some task >	So that I can < Achieve Some Goal >
1	User	Sign Up	I can save my details required for login
2	User	Login	I can get access to the system
3	User	Chang password	I can change my login credential
4	User	Add familiar person	Can add familiar person for face recognition
5	User	View familiar person	Can view the familiar person
6	User	Edit familiar person	Can edit the familiar person details
7	User	Delete familiar person	Can remove familiar person
8	Camera	Camera surveillance	Can identify visitors and familiar persons using face recognition
9	User	View Visitor logs	Can view Visitor logs entered by camera module
8	User	View device status	Can view device status
9	User	Update device status	Can update device status
10	Embedded Module	Update device status	Can update device status according to the instructions obtained

Figure 3.4: User Story

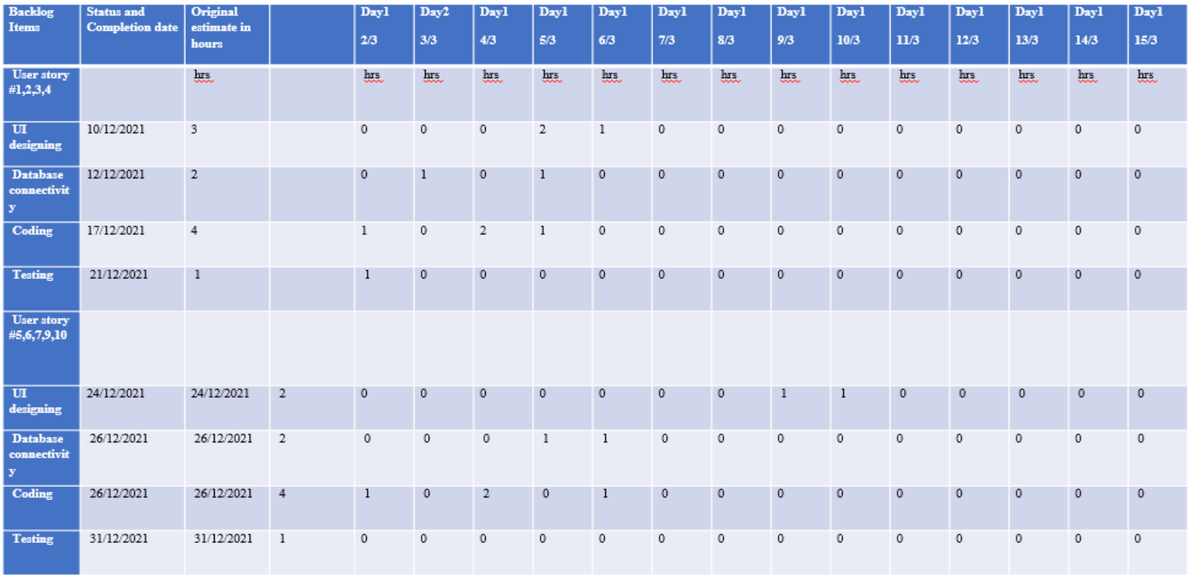
3.6.2 PROJECT PLAN



User Story ID	Task Name	Start Date	End Date	Days	Status
1	Sprint 1	01/12/2021	17/11/2021	2	Completed
2		08/12/2021	22/11/2021	3	Completed
3		15/12/2021	08/12/2021	5	Completed
4	Sprint 2	21/12/2021	22/12/2021	8	completed
5		30/12/2021	04/01/2022	5	completed
6	Sprint 3	05/01/2022	11/01/2022	6	completed
7		13/01/2022	22/01/2022	9	completed
8	Sprint 4	01/02/2022	09/02/2022	8	completed
9		15/02/2022	20/02/2022	5	completed

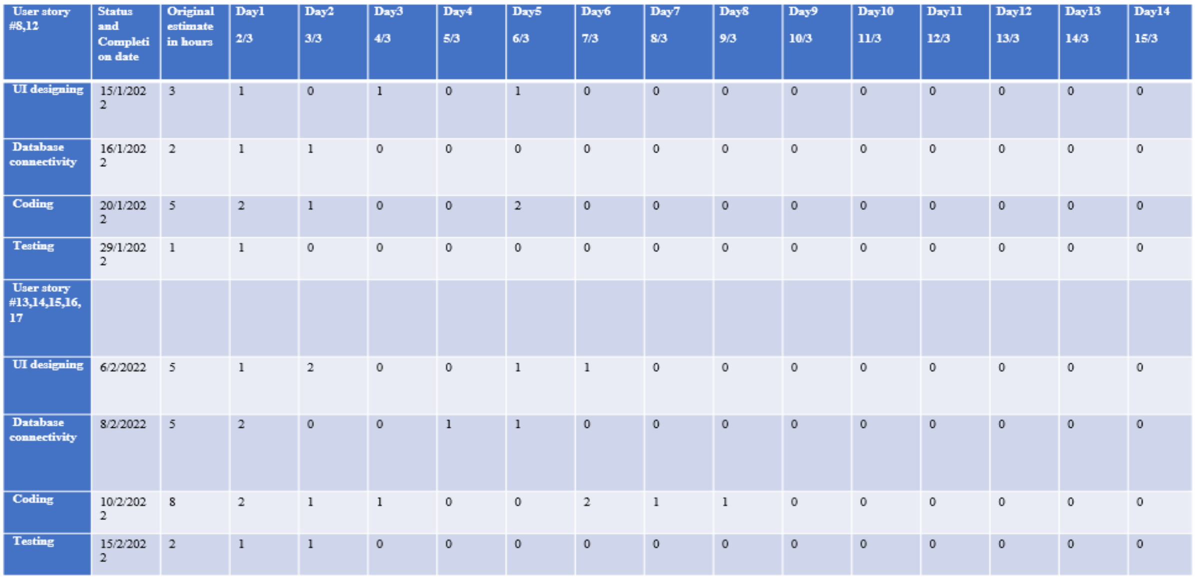
Figure 3.5: Project Plan

3.6.3 SPRINT BACKLOG PLAN



Backlog Items	Status and Completion date	Original estimate in hours		Day1 2/3	Day2 3/3	Day1 4/3	Day1 5/3	Day1 6/3	Day1 7/3	Day1 8/3	Day1 9/3	Day1 10/3	Day1 11/3	Day1 12/3	Day1 13/3	Day1 14/3	Day1 15/3
User story #1,2,3,4		hrs		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
UI designing	10/12/2021	3		0	0	0	2	1	0	0	0	0	0	0	0	0	0
Database connectivity	12/12/2021	2		0	1	0	1	0	0	0	0	0	0	0	0	0	0
Coding	17/12/2021	4		1	0	2	1	0	0	0	0	0	0	0	0	0	0
Testing	21/12/2021	1		1	0	0	0	0	0	0	0	0	0	0	0	0	0
User story #5,6,7,9,10																	
UI designing	24/12/2021	24/12/2021	2	0	0	0	0	0	0	0	1	1	0	0	0	0	0
Database connectivity	26/12/2021	26/12/2021	2	0	0	0	1	1	0	0	0	0	0	0	0	0	0
Coding	26/12/2021	26/12/2021	4	1	0	2	0	1	0	0	0	0	0	0	0	0	0
Testing	31/12/2021	31/12/2021	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

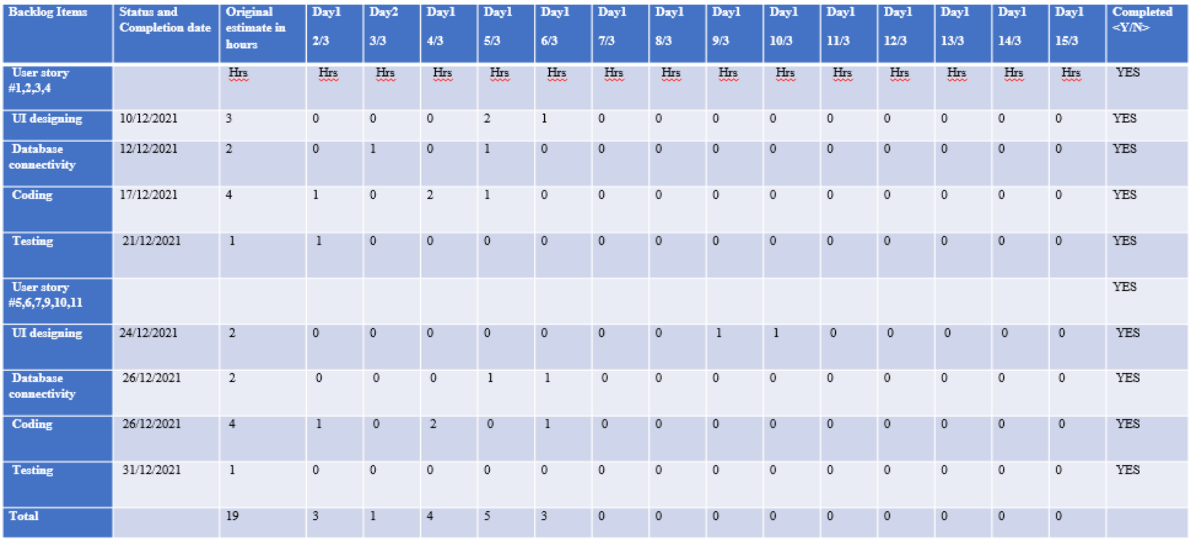
Figure 3.6: Sprint Backlog Plan 1.1



User story #8,12	Status and Completion date	Original estimate in hours	Day1 2/3	Day2 3/3	Day3 4/3	Day4 5/3	Day5 6/3	Day6 7/3	Day7 8/3	Day8 9/3	Day9 10/3	Day10 11/3	Day11 12/3	Day12 13/3	Day13 14/3	Day14 15/3
UI designing	15/1/2022	3	1	0	1	0	1	0	0	0	0	0	0	0	0	0
Database connectivity	16/1/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Coding	20/1/2022	5	2	1	0	0	2	0	0	0	0	0	0	0	0	0
Testing	29/1/2022	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
User story #13,14,15,16,17																
UI designing	6/2/2022	5	1	2	0	0	1	1	0	0	0	0	0	0	0	0
Database connectivity	8/2/2022	5	2	0	0	1	1	0	0	0	0	0	0	0	0	0
Coding	10/2/2022	8	2	1	1	0	0	2	1	1	0	0	0	0	0	0
Testing	15/2/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.7: Sprint Backlog Plan 1.2

3.6.4 SPRINT BACKLOG ACTUAL



Backlog Items	Status and Completion date	Original estimate in hours	Day1 2/3	Day2 3/3	Day1 4/3	Day1 5/3	Day1 6/3	Day1 7/3	Day1 8/3	Day1 9/3	Day1 10/3	Day1 11/3	Day1 12/3	Day1 13/3	Day1 14/3	Day1 15/3	Completed <Y/N>
User story #1,2,3,4		Hrs	Hrs	Hrs	Hrs	Hrs	Hrs	Hrs	Hrs	Hrs	Hrs	Hrs	Hrs	Hrs	Hrs	Hrs	YES
UI designing	10/12/2021	3	0	0	0	2	1	0	0	0	0	0	0	0	0	0	YES
Database connectivity	12/12/2021	2	0	1	0	1	0	0	0	0	0	0	0	0	0	0	YES
Coding	17/12/2021	4	1	0	2	1	0	0	0	0	0	0	0	0	0	0	YES
Testing	21/12/2021	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	YES
User story #5,6,7,9,10,11																	YES
UI designing	24/12/2021	2	0	0	0	0	0	0	0	1	1	0	0	0	0	0	YES
Database connectivity	26/12/2021	2	0	0	0	1	1	0	0	0	0	0	0	0	0	0	YES
Coding	26/12/2021	4	1	0	2	0	1	0	0	0	0	0	0	0	0	0	YES
Testing	31/12/2021	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	YES
Total		19	3	1	4	5	3	0	0	0	0	0	0	0	0	0	

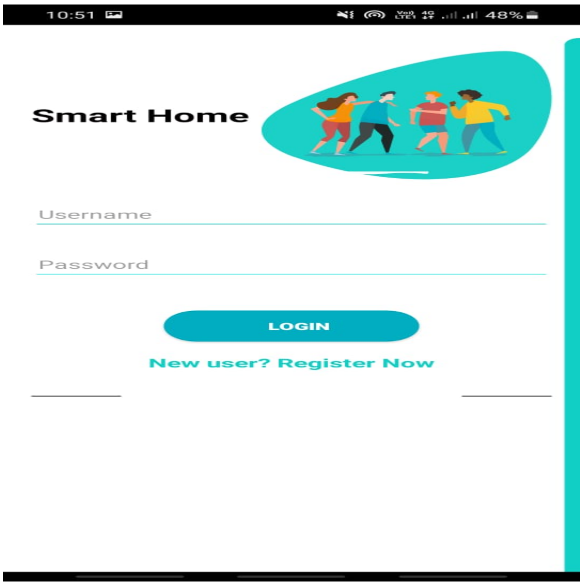
Figure 3.8: Sprint Backlog Actual

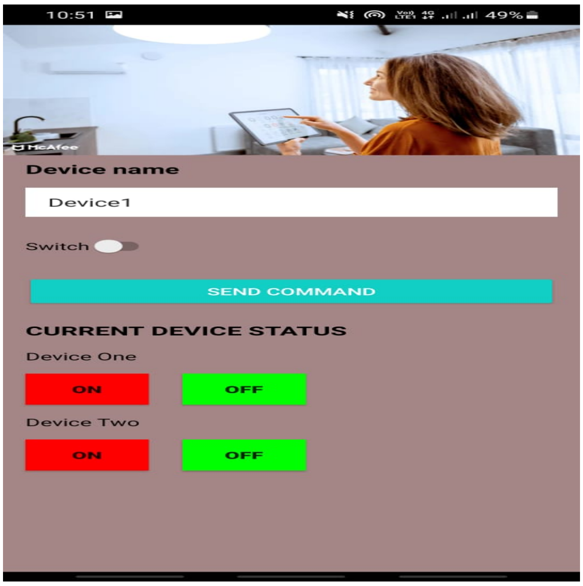
Chapter 4

Results and Discussions

4.1 Results

The experiment was carried out in Pentium iv 2.60GHz intel dual core processor, with 4 GB RAM, 15 LCD monitor with hard disk as 40 GB. The software required are Device control App, Arduino IDE, in windows operating system using C++ programming language. The resultant system was checked thoroughly by repeating the motion of opening the door multiple times to see if each time a notification is sent or not and by remotely switching the buzzer on or off from the Device control app which showed that the system works in the intended way and flawlessly. To test the endurance of the hardware, the setup was left turned on for a couple of hours and tested afterwards.





10:51 49%

Device name

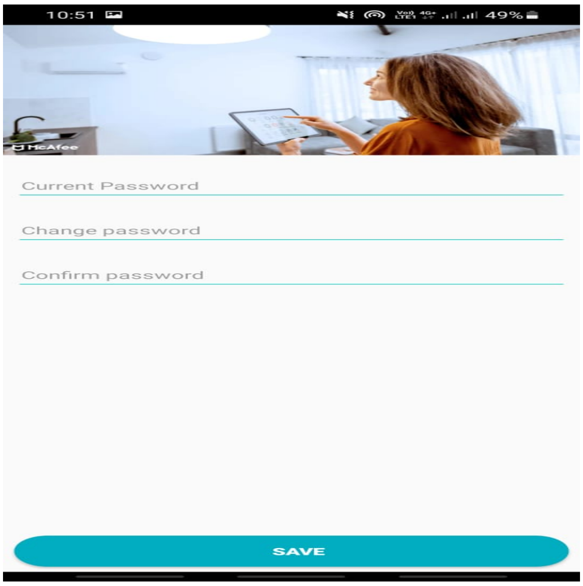
Device1

Switch

CURRENT DEVICE STATUS

Device One

Device Two

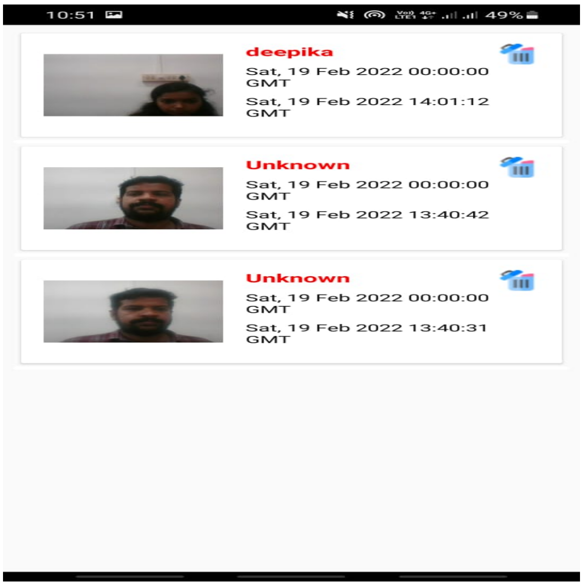


10:51 49%

Current Password

Change password

Confirm password

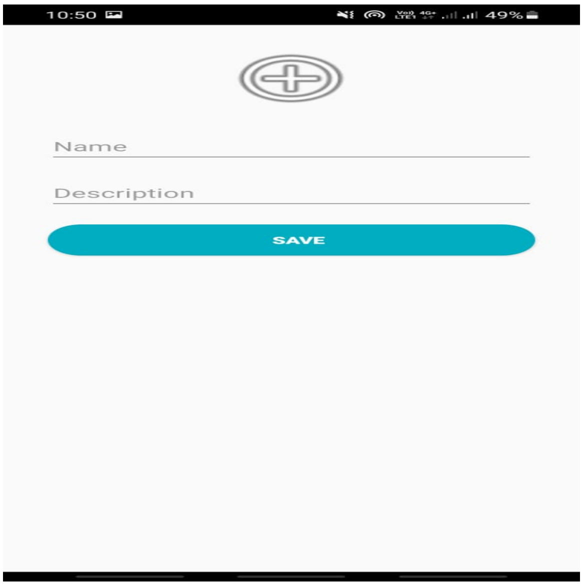


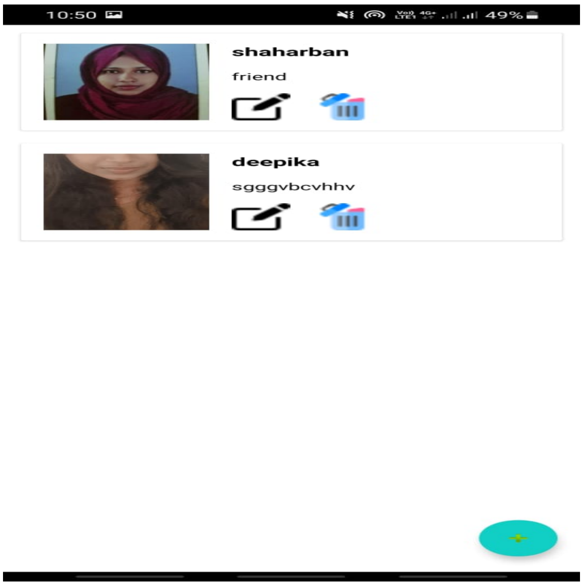
10:51 49%

deepika
Sat, 19 Feb 2022 00:00:00 GMT
Sat, 19 Feb 2022 14:01:12 GMT

Unknown
Sat, 19 Feb 2022 00:00:00 GMT
Sat, 19 Feb 2022 13:40:42 GMT

Unknown
Sat, 19 Feb 2022 00:00:00 GMT
Sat, 19 Feb 2022 13:40:31 GMT



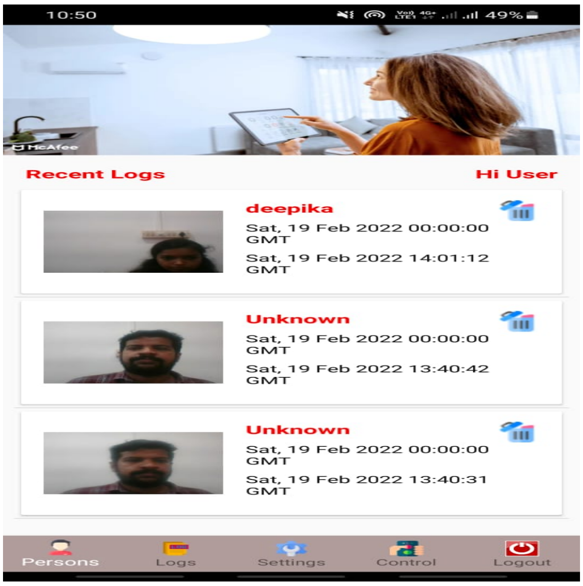


shaharban

friend

deepika

sgggvbcvhhv



Recent Logs

Hi User

deepika
Sat, 19 Feb 2022 00:00:00 GMT
Sat, 19 Feb 2022 14:01:12 GMT

Unknown
Sat, 19 Feb 2022 00:00:00 GMT
Sat, 19 Feb 2022 13:40:42 GMT

Unknown
Sat, 19 Feb 2022 00:00:00 GMT
Sat, 19 Feb 2022 13:40:31 GMT

Chapter 5

Conclusions

The proposed system sends the service request to concern person when sensor detects abnormality. In this project, we collect data from different sensors for monitoring the present status through IoT. This can be useful for intruder detection; fire accidents and any other abnormalities occur From the proposed system the authorized persons get the images captured by the camera via sms and the user will also get the count of people who visited the house.

References

[1] Ravi Kishore kodali and Vishal jain " IOT based smart security and Home Automation system" International conference on computing, communication and automation (ICCCA 2016)

[2] R. Piyare and M. Tazil, "Bluetooth based home automation system using cell phone," Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium on, Singapore, 2011, pp. 192-195

[3] S. Sen, S. Chakrabarty, R. Toshniwal, A. Bhaumik, "Design of an intelligent voice controlled home automation system", International Journal of Computer Applications, vol. 121, no.15, pp. 39-42, 2015

[4] H. AlShu'eili, G. S. Gupta and S. Mukhopadhyay, "Voice recognition based wireless home automation system," Mechatronics (ICOM), 2011 4th International Conference On, Kuala Lumpur, 2011, pp. 1-6

[5] R. Teymourzadeh, Salah Addin Ahmed, Kok Wai Chan and Mok Vee Hoong, "Smart GSM based Home Automation System," Systems, Process Control (ICSPC), 2013 IEEE Conference on, Kuala Lumpur, 2013, pp. 306-309.

[6] A. R. . C. Y. . O. K. Withanage, C., "A comparison of the popular home automation technologies," pp. 1 – 11, may 2014

Appendix

Source Code

```
# SMART_HOME_AUTOMATION.py

   from datetime import time

from flask import Flask,request,jsonify,session
from DBConnection import Db
app = Flask(__name__)
app.secret_key="kk"


@app.route(’/login’,methods=["post"])
def login():
   username = request.form[’username’]
   password = request.form[’password’]

   qry="SELECT ‘lid‘ FROM ‘login‘ WHERE ‘username‘=’"+username+"’ AND ‘password‘=’"+password+"’"
   d=Db()
   res=d.selectOne(qry)
   if res is not None:
      return jsonify(status=’ok’,lid=res["lid"])
   else:
      return jsonify(status=’no’)

@app.route(’/change_password’,methods=["post"])
def change_password():
   currentpassword = request.form[’currentpassword’]
   newpassword = request.form[’newpassword’]
   confirmpassword = request.form[’confirmpassword’]
   ps=request.form["psw"]
   lid=request.form["lid"]
   if currentpassword==ps:
      if confirmpassword==newpassword:
         qry="UPDATE ‘login‘ SET PASSWORD=’"+confirmpassword+"’WHERE ‘lid‘=’"+lid+"’"
         d=Db()
         res=d.update(qry)
         return jsonify(status=’ok’)
      else:
         return jsonify(status=’no’)
   else:
      return jsonify(status=’no’)


@app.route(’/add_familiar_person’,methods=["post"])
def add_familiar_person():
   name = request.form[’name’]
   description = request.form[’description’]
   ulid = request.form[’lid’]

   image = request.form[’image’]
   import datetime
   import base64
   timestr = datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
   print(timestr)
   a = base64.b64decode(image)
   fh = open("E:\\Final\\SMART_HOME_AUTOMATION\\SMART_HOME_AUTOMATION\\static\\familiarperson\\" + timestr + ".jpg", "wb")
   path = "/static/familiarperson/" + timestr + ".jpg"
   fh.write(a)
   fh.close()

   qry="INSERT INTO ‘familiar‘(‘name‘,‘photo‘,‘description‘,lid)VALUES(’"+name+"’,’"+path+"’,’"+description+"’,’"+ulid+"’)"
   d=Db()
   res=d.insert(qry)
   return jsonify(status=’ok’)

@app.route(’/device_control’,methods=["post"])
def device_control():
   deviceid=request.form[’deviceid’]
   if deviceid=="Device1":
      deviceid="1"
   else:
      deviceid="2"



   status=request.form[’status’]
   qry="INSERT INTO ‘devicerequest‘ (‘deviceid‘,‘command‘) VALUES (’"+deviceid+"’,’"+status+"’)"
   print(qry)
   d=Db()
   res=d.insert(qry)
   return jsonify(status=’ok’)

@app.route(’/view_familiar_person’,methods=["post"])
def view_familiar_person():
   lid=request.form["lid"]
   qry="SELECT * FROM ‘familiar‘ where lid=’"+lid+"’"
   d=Db()
   res=d.select(qry)
   return jsonify(status=’ok’,users=res)

@app.route(’/view_visitor_log’,methods=["post"])
def view_visitor_log():
   qry="SELECT * FROM ‘visitor log‘ order by ‘vlog id‘ DESC"
   d=Db()
   res=d.select(qry)
   return jsonify(status=’ok’,data=res)

@app.route(’/delete_familiar_person’,methods=["post"])
def delete_familiar_person():
   fid=request.form[’fid’]
   qry="DELETE FROM ‘familiar‘ WHERE fid=’"+fid+"’"
   d=Db()
   res=d.delete(qry)
   return jsonify(status=’ok’)
```

Appendix

```
@app.route('/edit_familiar_person',methods=["post"])
def edit_familiar_person():
   name = request.form['name']
   description = request.form['description']
   fid = request.form['fmid']

   image = request.form['image']
   if image!="0":
      import datetime
      import base64
      timestr = datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
      print(timestr)
      a = base64.b64decode(image)
      fh = open("C:\\Users\\user\\PycharmProjects\\SMART_HOME_AUTOMATION\\static\\familiarperson\\" + timestr + ".jpg",
             "wb")
      path = "/static/familiarperson/" + timestr + ".jpg"
      fh.write(a)
      fh.close()

      qry = "UPDATE `familiar` SET name='"+name+"',photo='"+path+"',description='"+description+"' WHERE `fid`='"+fid+"'"
      d = Db()
      res = d.update(qry)
      return jsonify(status='ok')
   else:
      qry = "UPDATE `familiar` SET name='" +name+"',description='" +description+"' WHERE `fid`='" + fid + "'"
      d = Db()
      res = d.update(qry)
      return jsonify(status="ok")

@app.route('/user',methods=["post"])
def user():
   username=request.form['username']
   password=request.form['password']
   place=request.form['place']
   district=request.form['district']
   email=request.form['email']
   image=request.form['image']
   import datetime
   import base64
   timestr = datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
   print(timestr)
   a = base64.b64decode(image)
   fh = open("E:\\Final\\SMART_HOME_AUTOMATION\\SMART_HOME_AUTOMATION\\static\\userpics\\" + timestr + ".jpg", "wb")
   path = "/static/userpics/" + timestr + ".jpg"
   fh.write(a)
   fh.close()
   phone=request.form['phone']
   qry1 = "INSERT INTO `login`(`username`,`password`)VALUES('" + username + "','" + password + "')"
   d = Db()
   print(qry1)
   lid = str(d.insert(qry1))
   qry="INSERT INTO
        `user`(`uname`,`place`,`district`,`email`,`image`,`phone`,`lid`)VALUES('"+username+"','"+place+"','"+district+"','"+email+"','"+path+"
   d=Db()
   res=d.insert(qry)
   return jsonify(status='ok')




@app.route('/view_visitor_notification',methods=["post"])
def view_visitor_notification():
   id=request.form['idd']
```

```
    print(id)
    qry = "SELECT `visitor log`.*,`visitor log`.`vlog id`as vid FROM `visitor log` WHERE `vlog id`>'"+id+"' ORDER by `vlog
        id` DESC "
    d = Db()
    res = d.selectOne(qry)
    if res is not None:
        return jsonify(status='ok', nid=res['vid'],image=res['photo'],time=res['time'],date=res['date'])
    else:
        return jsonify(status="no")


@app.route('/delete_visitor_log',methods=["post"])
def delete_visitor_log():
    vid=request.form['vid']
    qry="DELETE FROM `visitor log`WHERE `vlog id`='"+vid+"'"
    d=Db()
    res=d.delete(qry)
    return jsonify(status='ok')


#
# @app.route('/owner_viewdevicestatus',methods=['post'])
# def owner_viewdevicestatus():
#     db=Db()
#     qry="SELECT status FROM `device control` WHERE `device id`='1'"
#     res1=db.selectOne(qry)
#     status1= res1['status']
#
#     qry = "SELECT status FROM `device control` WHERE `device id`='2'"
#     res2=db.selectOne(qry)
#     status2 = res2['status']
#
#     return jsonify(status="ok", status1=status1,status2=status2)
#
#



@app.route('/owner_viewdevicestatus',methods=['post'])
def owner_viewdevicestatus():
    db=Db()
    qry="SELECT status FROM `device control` WHERE `device id`='1'"
    res1=db.selectOne(qry)
    status1= res1['status']

    qry = "SELECT status FROM `device control` WHERE `device id`='2'"
    res2=db.selectOne(qry)
    status2 = res2['status']

    return jsonify(status="ok", status1=status1,status2=status2)


@app.route('/getlaststaus',methods=['post'])
def getlaststatus():
    id=request.form['id']
    qry="SELECT * FROM `devicerequest` WHERE `devicereqid` >"+id +" order by devicereqid "
    print(qry)
    db=Db()
    res=db.select(qry)
    if len(res)>0:
        a=res[0]
        print("aaaaa", a['deviceid'] ,",,," , a['command'])
        return jsonify(status="ok",devid= a['deviceid'], val=a['command'],id=a['devicereqid'])
    else:
        return jsonify(status="no")



@app.route('/updatedevicestatus',methods=['post'])
def updatedevicestatus():
    device1=request.form["device1"]
    device2=request.form["device2"]


    print("Device status" , device1,"====",device2)

    if device1=="1":
        device1="on"
    else:
        device1="off"

    if device2=="1":
        device2="on"
    else:
        device2="off"

    print(device1,device2,"888888888")

    db=Db()
    db.update("UPDATE `device control` SET STATUS='"+device1+"' WHERE `device id`='1'")
    db.update("UPDATE `device control` SET STATUS='"+device2+"' WHERE `device id`='2'")

    return jsonify(status="ok")
if __name__ == '__main__':
    app.run(debug=True,host='0.0.0.0')
```

Appendix

Arduino Uno program Code

```
const int led1 = 2;
const int led2 = 3;
const int led3=4;
const int led4=5;

int led1_status =0,led1_flag,led2_status =0,led2_flag,led3_status =0 ,led3_flag,led4_status =0,led4_flag;

int send_count=0;
int send_flag=0;
int i=0;
char rec_data;
void setup()
{
  // put your setup code here, to run once:
 pinMode(led1, OUTPUT);
 pinMode(led2, OUTPUT);
 pinMode(led3, OUTPUT);
 pinMode(led4, OUTPUT);

 Serial.begin(9600);

}

void loop()
{

   if (Serial.available() > 0) {
    // read the incoming byte:
    rec_data = Serial.read();

    // say what you got:


  }


  send_count=send_count+1;


  if (send_count>70)
  {


    send_count=0;
    send_flag=1;

  }


  if (rec_data==’A’ && led1_status ==0)
  {
    digitalWrite(led1,HIGH);
    led1_status = 1;
    delay(500);
    rec_data=’0’;
  }
    if (rec_data==’B’ && led1_status == 1 )
  {
    digitalWrite(led1,LOW);
    led1_status = 0;
    delay(500);
    rec_data='0';
  }

  if (rec_data=='C' && led2_status ==0)
  {
    digitalWrite(led2,HIGH);
    led2_status = 1;
    delay(500);
    rec_data='0';
  }
    if (rec_data=='D' && led2_status == 1 )
  {
    digitalWrite(led2,LOW);
    led2_status = 0;
    delay(500);
    rec_data='0';
  }


    if (send_flag==1)
    {
     delay(50);
     Serial.print('*');
     delay(50);
     Serial.print(led1_status);
     delay(50);
     Serial.print(',');
     delay(50);
     Serial.print(led2_status);
     delay(50);


     Serial.print('#');
     delay(50);
     send_flag=0;

   }

 }
```

Database Design

Attribute Name	Datatype	Width	Description
Lid	Integer	11	Primary Key
Username	Varchar	50	Not Null

Table A.1: Login

Attribute Name	Datatype	Width	Description
Fid	Integer	11	Primary Key
Name	Varchar	50	Not Null
Photo	Varchar	50	Not Null
Description	Varchar	50	Not Null

Table A.2: Familiar

Attribute Name	Datatype	Width	Description
Deviceid	Integer	11	Primary Key
Status	Varchar	50	Not Null

Table A.3: Device control

Attribute Name	Datatype	Width	Description
Vlogid	Integer	11	Primary Key
Date	Date	20	Not Null
Time	Time	20	Not Null
Visitor Name	Varchar	20	Not Null
Photo	Varchar	50	Not Null

Table A.4: Visitor log

Attribute Name	Datatype	Width	Description
Nid	Integer	11	Primary Key
Vlogid	Integer0	11	Foreign Key
Date	Date	20	Not Null
Status	Varchar	50	Not Null

Table A.5: Notification

Attribute Name	Datatype	Width	Description
Uid	Integer	11	Primary Key
Uname	Varchar	50	Not Null
Place	Varchar	50	Not Null
District	Varchar	50	Not Null
Email	Varchar	50	Not Null
Image	Varchar	50	Not Null
Phone	Bigint	20	Not Null
Lid	Integer	50	Foriegn Key

Table A.6: User registration

Dataflow Diagram

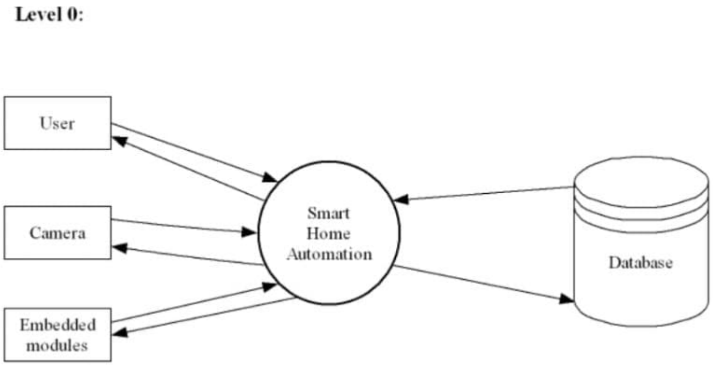


Figure A.1: Dataflow Diagram

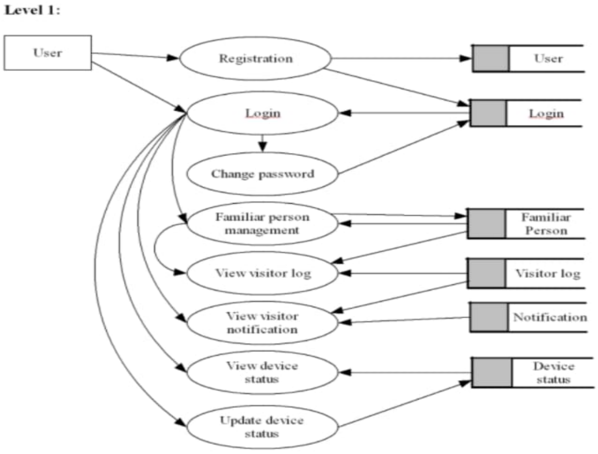


Figure A.2: Dataflow Diagram

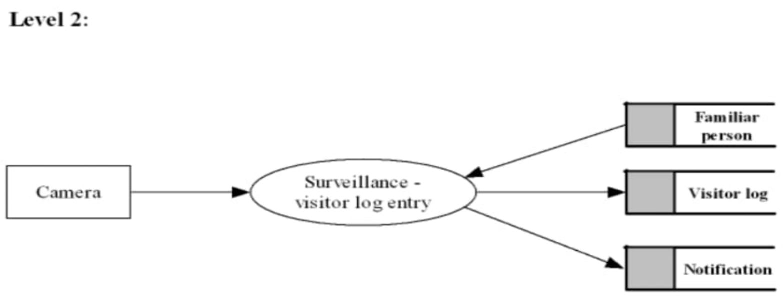


Figure A.3: Dataflow Diagram

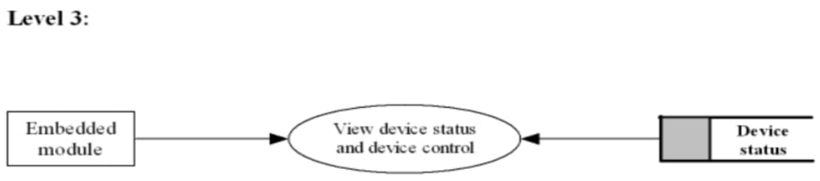


Figure A.4: Dataflow Diagram